LOCKHEED MARTIN

Open Systems Joint Task Force

Uniform Driver Interface (UDI)

Open Systems Project Engineering Conference (OSPEC)

FY 98 Status Review

29 April - 1 May 1998

Curtis Royster

Defense Information Systems Agency (DISA)

Mark Evenson/Larry Anderson

Hewlett Parkard &

Lockheed Martin Tactical Defense Systems

Background

GENERAL PROBLEM:

- Device driver code is not portable across platforms, interconnects, or operating systems.
- No open standards for device driver interface to OS and/or hardware.
- A new device driver must be written (for one device) for each new target OS or platform using that device.
- Device driver code is complex.

OS-JTF UNIFORM DRIVER INTERFACE (UDI) PARTICIPATION:

Lockheed-Martin will:

- Identify real-time I/O driver requirements from a variety of weapon systems and compare to UDI Environment draft specs.
- Build a prototype driver to demonstrate suitability of proposed UDI specification for a SCI network interface.

OPEN SYSTEMS STANDARDIZATION

- **A POSIX UDI STUDY GROUP WAS FORMED IN OCT 95**
  - **Hewlett Parkard, Digital, IBM, NCR, SCO, Adaptec, Interphase, & Lockheed Martin**
- **DRAFTED A PAR IN JAN 96**
- **SUSPENDED ACTIVITY IN MAR 96 (PAR NOT SUBMITTED)**
  - **Mature existing practice first (prototype, industry acceptance)**
  - **Pursue fast track to standardization once technology mature**
- **RESUMPTION OF STANDARDIZATION ACTIVITIES PLANNED (X-OPEN)**

LOCKHEED MARTIN

Open Systems OS Joint Task Force

What is UDI

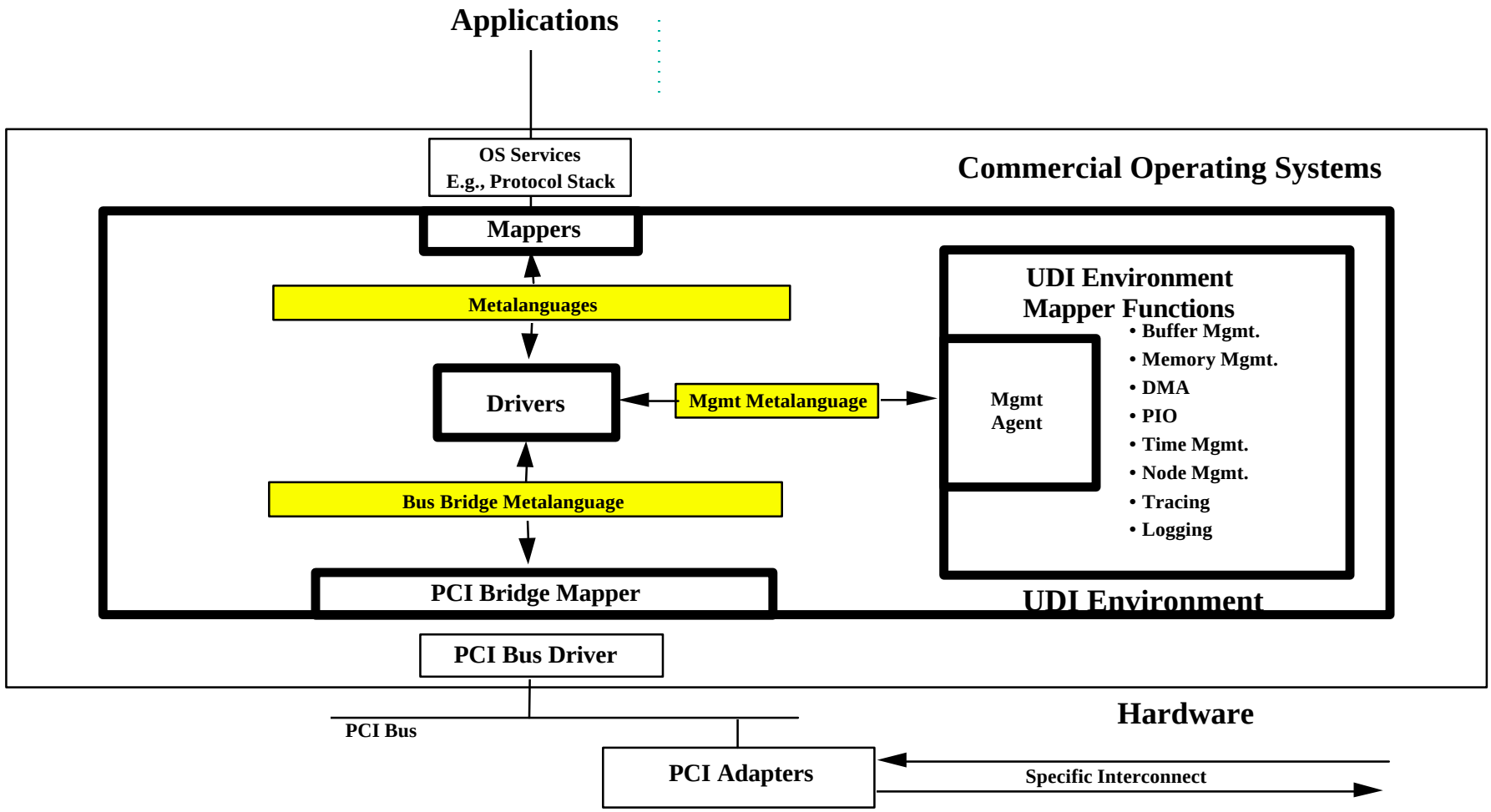
➔ **Complete set of driver services; every thing a device driver needs to:**

- **Access and control its device,**
- **Get configured into the system, and**
- **Communicate with other drivers or system modules**

➔ **No OS or platform-specific components in the driver**

➔ **No OS policy in the driver**

UDI Prototype Environment



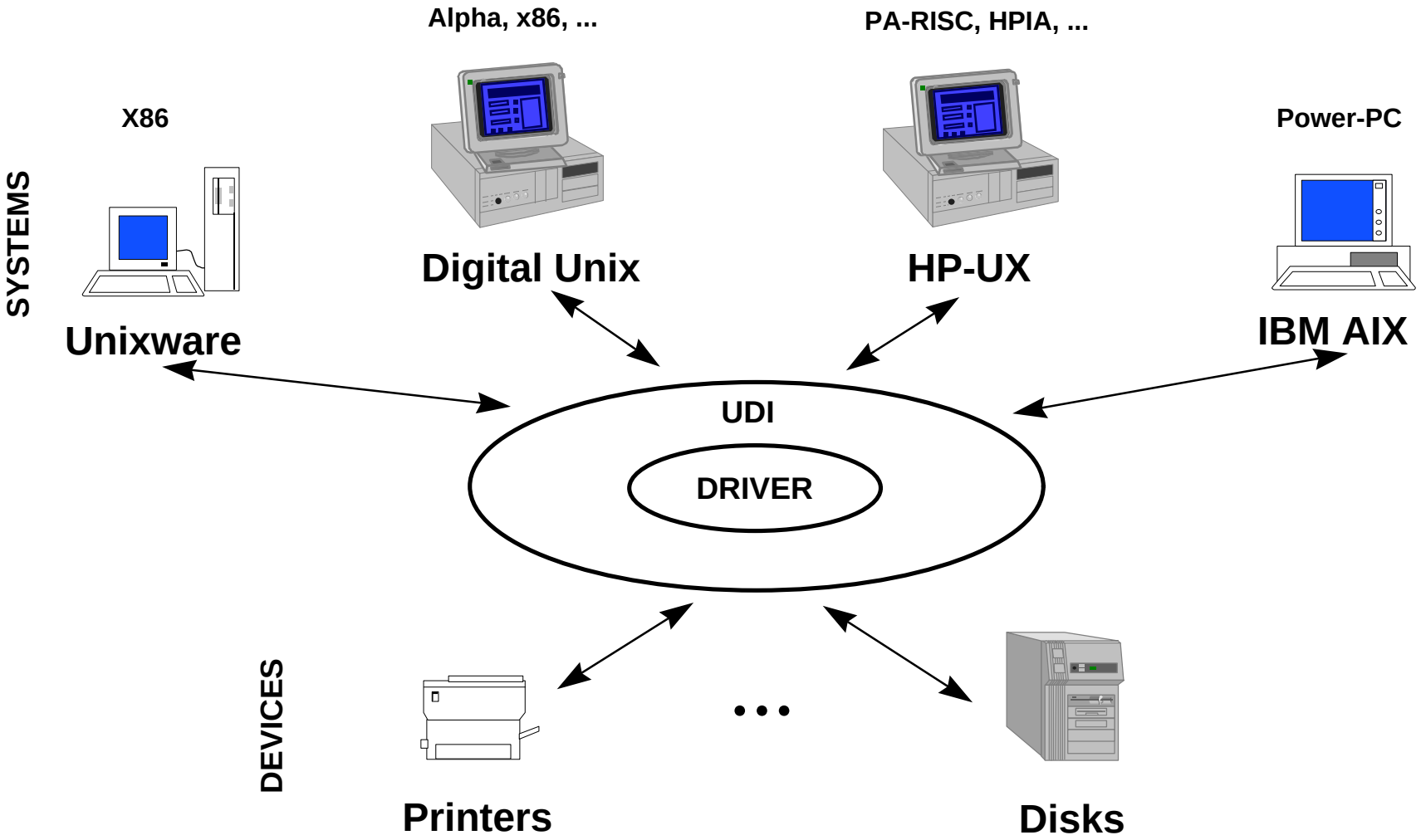
LOCKHEED MARTIN

Open Systems OS Joint Task Force

UDI Accomplishments

- Draft 0.75 Specification Release (10/96)
  - Includes UDI services
  - – 5 metalanguages (management, bus/bridge, network, SCSI, pointer)
  - – Basis for a “Proof of Concept” prototype implementation
- Proof of Concept Prototype Demo (12/97)
  - Portable SCSI & network drivers on 4 different platforms & 5 OSs
  - Custom real-time networking application over UDI
- 1.0 Specification Finalization In Progress
  - Incorporate “lessons learned” from the prototyping effort
  - Complete miscellaneous requirements (hot plug support, tracing & logging, packaging & distribution, branding)

UDI Participants and Interface Illustrations



LOCKHEED MARTIN

UDI Program Future Plans

- Complete 1.0 Specification
  - Update based on Prototype
  - Resolve open issues
    - Performance/Metalanguage Issues
  - Add Missing Features
    - Address assignment between bus/bridge devices
    - Support addresses > 32 bits
- Broaden Implementation
  - Technology spread (e.g., I2O compatible)
  - Support embedded systems (e.g., VXWorks, Rational, ...)
- Re-Apply to Formal Standards Body (Following 1.0 Spec. Completion)
- Define Certification (Driver/UDI Environment Version Compatibility)
- Productization (e.g., UDI Demonstration - UniForum 8/98)

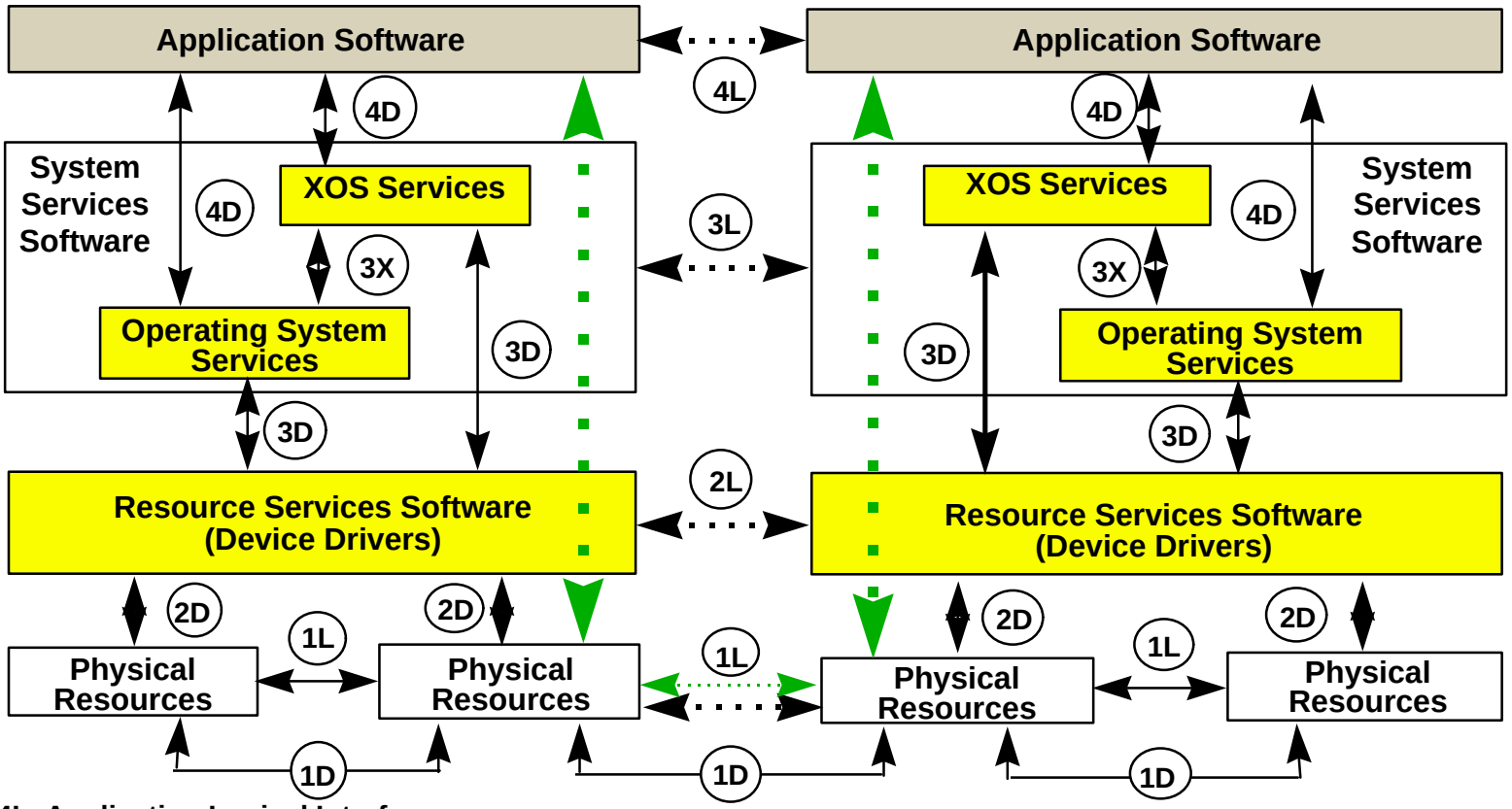
LOCKHEED MARTIN

Open Systems OS Joint Task Force

UDI & I2O

- Started in similar timeframes (I2O at end of '94; UDI Specification work began in Nov. '94)
- Complementary technologies; UDI driver services can be used to support I2O
  - I2O OSMs
  - UDI drivers/services on the IOP
- Different focus
  - I2O is focused on a particular hardware architecture (a host and an IOP)
  - UDI is focused solely on driver commonality

Generic Open Architecture (GOA)



4L Application Logical Interfaces
4D Application Direct Interfaces (e.g., APIs)
3L System Service Logical Interfaces
3D OS/EXtended Service Direct Interfaces
3X EXtended/OS Interfaces

2L Device Driver Logical Interfaces
2D Device Driver/Physical Direct Interfaces
1L Physical/Physical Logical Interfaces
1D Physical/Physical Direct Interfaces

LOCKHEED MARTIN

OS-JTF SOW Summary

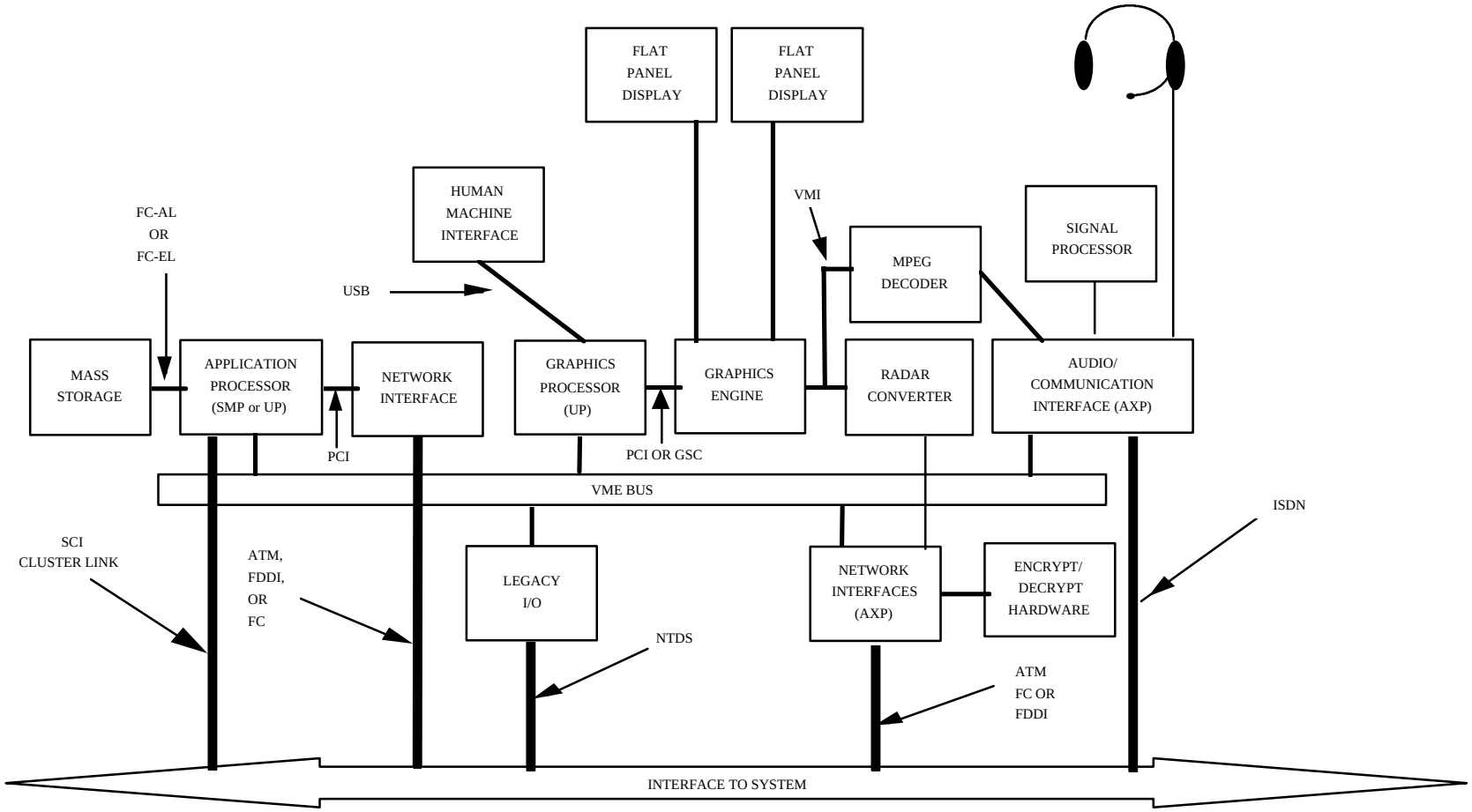
Issue: Will UDI Environment meet military system I/O interconnect protocol and performance requirements?

Subtasks:

- ➔ **Document UDI Related Military System I/O Requirements**
  - **Create Requirements Template**
  - **Collect/Document Military System Requirements**

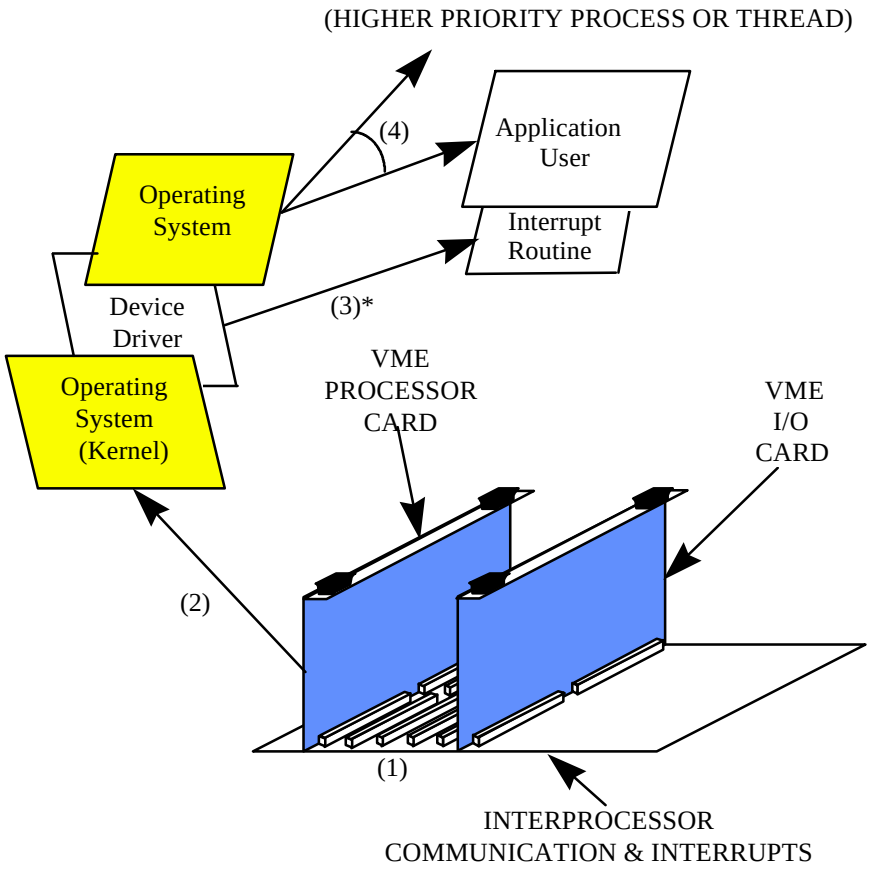
  (ASUW Improvement Program - AIP, Light Airborne Multi-Purpose System Helicopter - LAMPS, Advanced Q-70 Display System, and C-130 Combat Talon II System)
  - **Draft UDI Benchmark Document**
- ➔ **Implement Prototype UDI Environment Capabilities**
  - **Create Prototype SCI Metalanguage for SCI adapters/drivers**
  - **Implement Prototype PCI/SCI Driver**
  - **Draft Preliminary Standard SCI Metalanguage Specification**

Advanced Display System (Q-70)



SMP=SYMMETRIC MULTIPROCESSOR UP=UNIT PROCESSOR AXP=AUXILARY PROCESSOR

Advanced Display System (Q-70)



Operational Need: The Aegis Advanced Computing Environment (ACE) Specification defines critical Commercial-Off-The-Shelf (COTS) compatible performance parameters for the Q-70 architecture.

Requirement: Interrupt and associated latencies.

Allocation Times (See Figure):

1. **Point 1 - Inter-processor Communication (40 Mbytes/second**
2. **Point 2 - Interrupt Response Latency (100usec.)**
3. **Point 3 - Process Dispatch Latency (250 usec. *)**
4. **Point 4 - Kernel Preemption Latency 100 usec.)**

*** Worst case time includes the interrupt response latency, execution of device driver or kernel code servicing the interrupt, overhead required in context switching, and any critical sections when context switching is disabled by operating system.**

LOCKHEED MARTIN

SCI Metalanguage Definition

- Models IEEE P1596.9 Physical Layer API for Scalable Coherent Interface (SCI PHY-API)
- SCI PHY-API is an implementation independent physical interface for SCI applications (IEEE 1596.9)

SCI Metalanguage & P1596.9 Functional Area Commonality

- Configuration support (driver, device, network)
- Exception handling & reporting
- Memory address translation/mapping support
- Transparent and non-transparent memory-to-memory transactions via PIO
- Block data transfer via DMA
- IEEE 1212 message passing
- IEEE 1212 remote interrupt receipt
- User constructed SCI packet transfer
- Cache (line) transactions

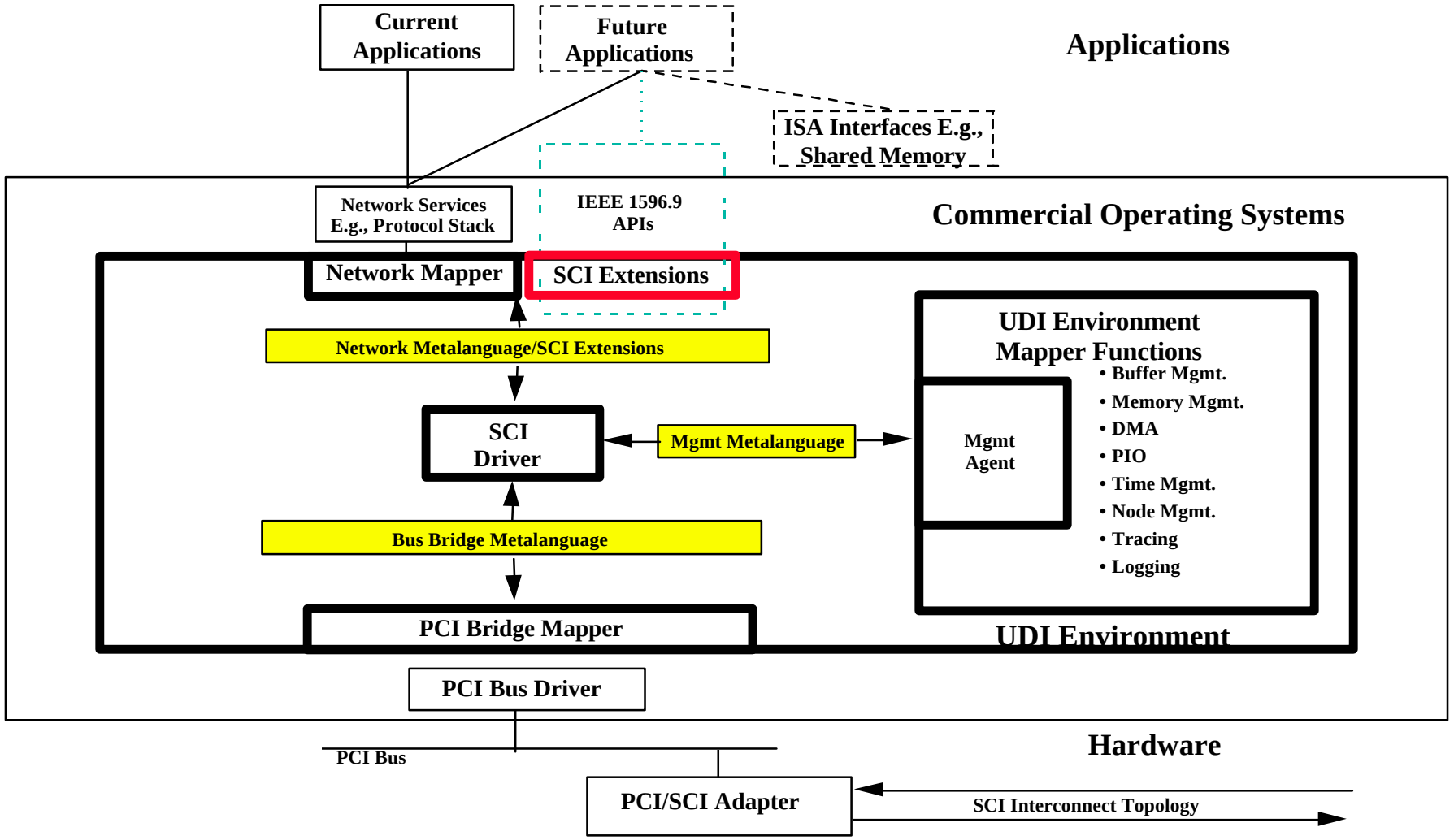
LOCKHEED MARTIN

OS-JTF Metalanguage Contribution

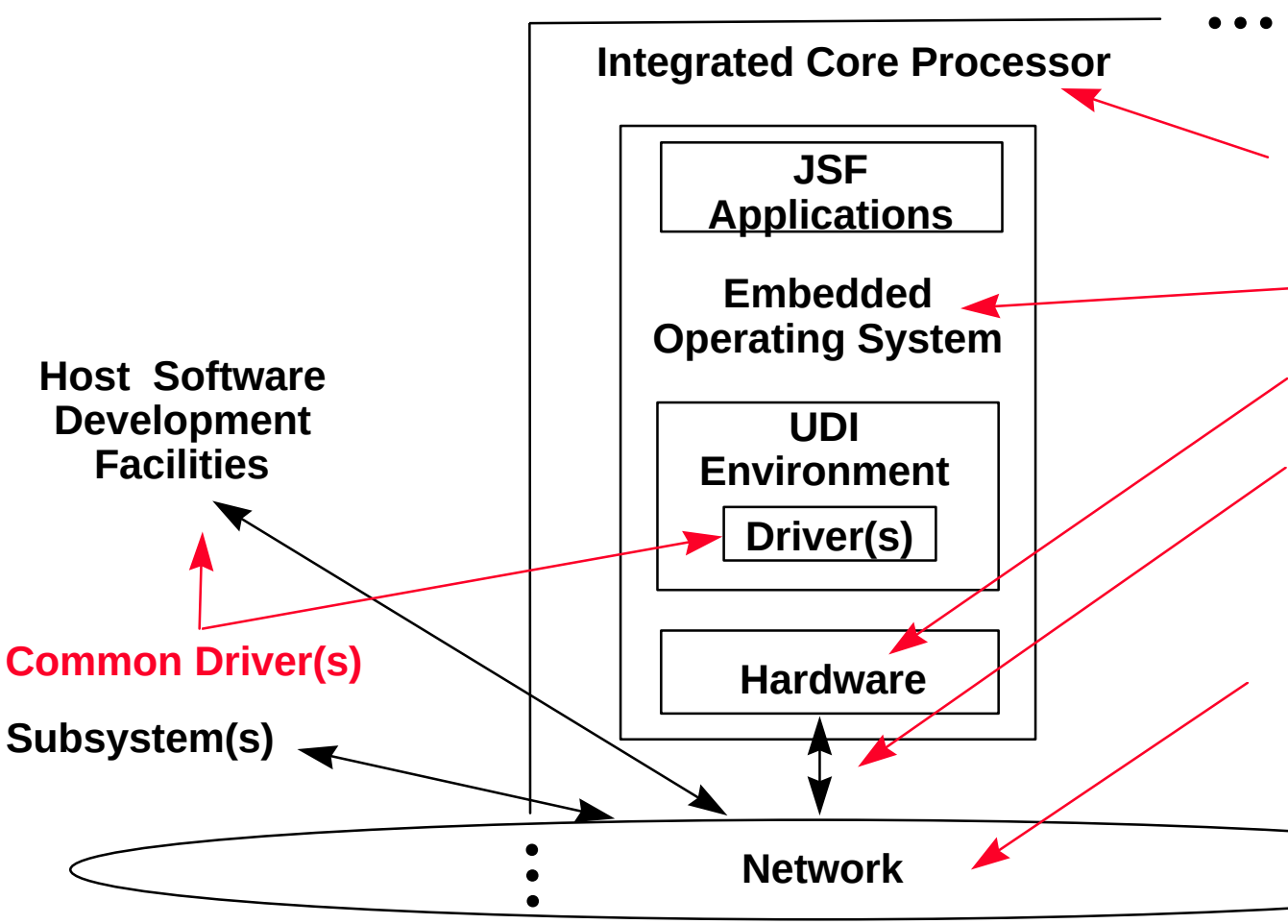
UDI Model and Metalanguage Issues

- ➔ UDI employs *messaging* for communication between driver and environmental layers
  - Supports split-driver model implementation
  - I/O service's client may reside in a different domain than the I/O driver
  - Compatible with Input/Output Processors (e.g., I2O IOP model)
- ➔ SCI employs *direct memory access* across distributed memory
  - Utilizes memory mapped I/O model
  - Applications may by-pass OS service call/interrupt processing overhead
  - Compatible with lightweight communication protocols like VIA
- ➔ Metalanguage Issues
  - Address assignment between bus/bridge devices (UDI Spec. change pending)
  - Address support > 32 bits (deferred until after UDI Spec. 1.0)

UDI Prototype Environment



Defense Program Flexibility (JSF Illustration)



Options

Architecture
- Message Passing
- Shared Memory
- Cache Coherent
- Heterogeneous

Embedded OSs
- VXWorks, Rational, ...

Adapter
- PCI, VIA, NUMA, ...

Interface Protocol
- Fibre Channel
- SCI
- Firewire
- etc

Network Topology
- Unified
- Scalable
- Heterogeneous

LOCKHEED MARTIN

UDI Information

Web Page

http://www.sco.com/UDI/

FTP Site

ftp://telford.nsa.hp.com/pub/hp_stds/udi

Project UDI Contacts

Chair: Kevin Quick, +1 214 654 5173. kquick@iphase.com

Vice Chair: Mark Evenson, +1 408 447 5601, mevenson@cup.hp.com

Secretary: John Lee, +1 415 786 5323, john.lee@eng.sun.com

Editor: Kurt Gollhardt, +1 908 790 2277, kdg@sco.com

Reference Material/Access

Project Deliverables

- UDI Environment Interface Requirements Specification
- UDI SCI Metalanguage Specification (UDI Specification Input)

Standardization Documentation

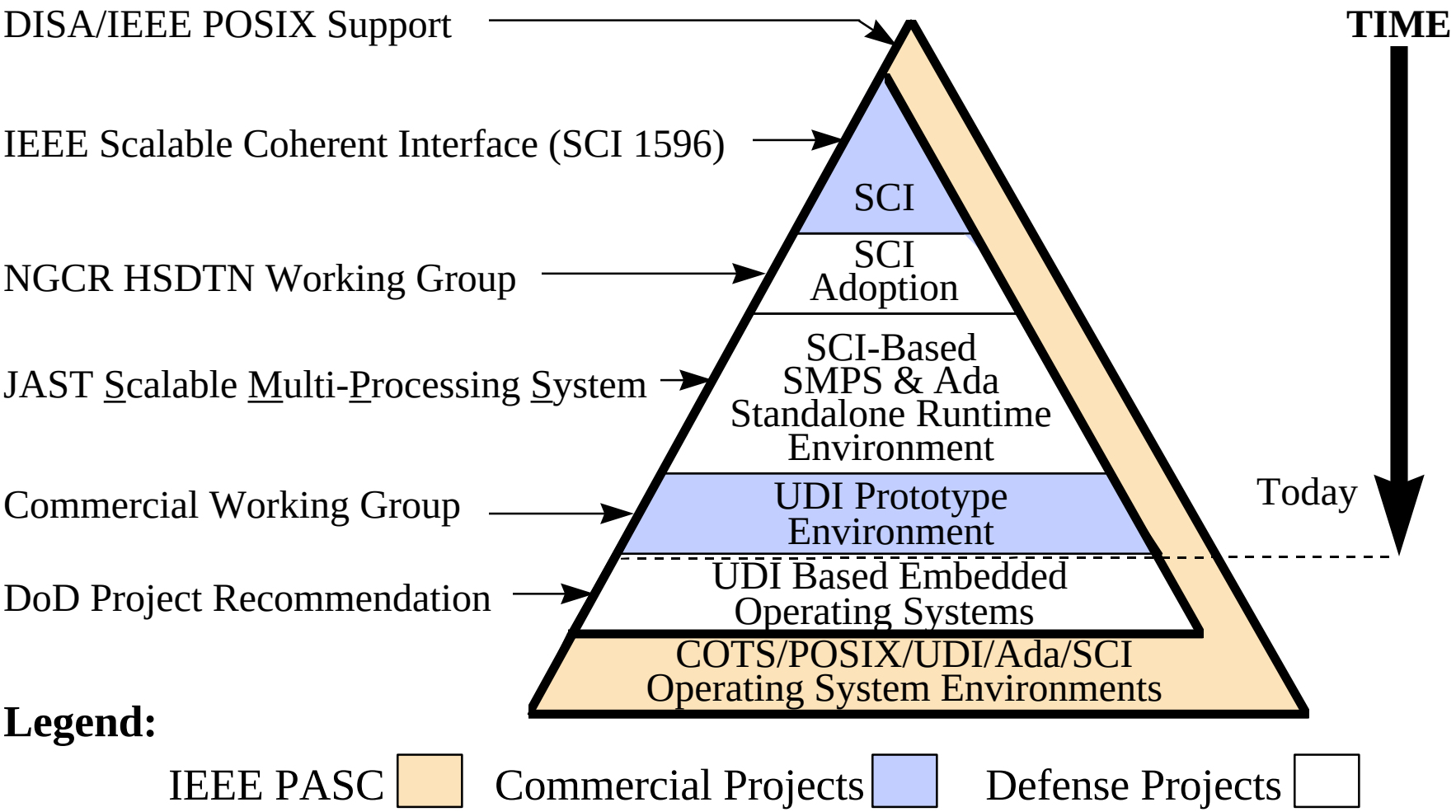
- Draft UDI Specification (Document Set)
  - ⇒ http://www.sco.com/UDI/
- Physical Layer API for Scalable Coherent Interface (SCI)
  - ⇒ http://sci.lbl.gov/
- Standard for Scalable Coherent Interface (SCI-IEEE 1596)
  - ⇒ To order call 1-800-678-IEEE (US)

LOCKHEED MARTIN

Open Systems OS Joint Task Force

Backup

Open System History & Recommendations



LOCKHEED MARTIN

SCI PHY-API Purpose
(From Presentation by IEEE 1596.9 Chairman)

- Define standard SCI physical layer Application Program Interface (API)
- Hide implementation specifics and implementation scope within standard driver without unnecessary overhead in case of full (SCI) implementation
- Guarantee interchangeability of SCI hardware at its front-end (SCI) and back-end (driver API)
- Allow SCI application software to be developed completely hardware independent

LOCKHEED MARTIN

Transaction Groups
(From Presentation by IEEE 1596.9 Chairman)

Block Transactions

- Always asynchronous
- Calls private CallBack procedure upon completion
- Variable transaction size e[0,215] bytes
- Procedure call

Selected Byte Transactions

- Synchronous for reads; asynchronous for posted writes
- Calls global error handler or private CallBack procedure depending on posting flags upon completion
- Variable transaction size e[0,16] bytes
- Procedure call

Transparent Transactions

- Synchronous for reads; asynchronous for posted writes
- Calls global error handler or private CallBack procedure depending on posting flags upon completion
- Fixed transaction size e[0,16] bytes
- Procedure call or assignment statement *

* Dependent on SCI interface hardware available

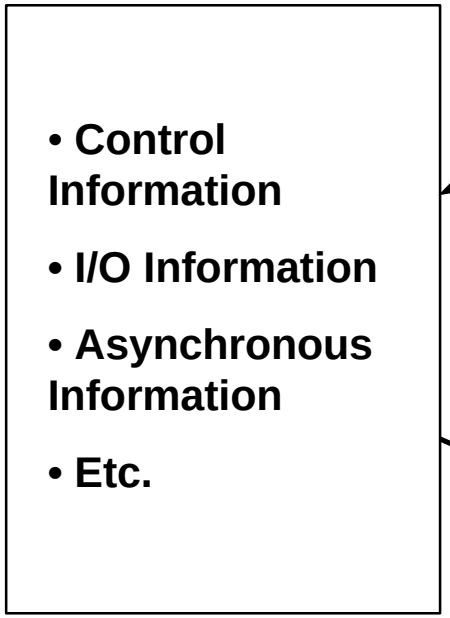
LOCKHEED MARTIN

Open Systems OS Joint Task Force

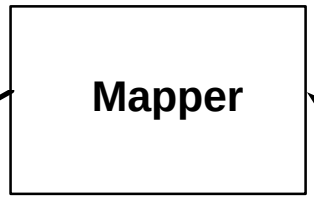
Prototype UDI Accomplishments

- UDI Standard Working Group Participation
  - Participation in weekly UDI Technical Group teleconferences
  - Attendance at two UDI Working Group planning meetings
  - Participation in weekly UDI demonstration planning telcons
- SCI Metalanguage Definition (UDI Specification Draft Input)
  - Compatible with draft IEEE 1596.9 Working Group specification
  - Submitted to UDI Environment Working Group
- UDI Prototype Driver Development
  - Uses Dolphin Interconnect Solutions PCI/SCI Adapter
  - Coding approximately 95% complete; ready for debug
  - Employs UDI Environment services; management, bus/bridge, and network metalanguages/mappers

Metalanguage Model



Control Block
(Memory Resource)



Multiple Entry Points

Request

Metalanguage Code

Response

Responsibility

- Metalanguage Linkage (Channels, e.g.:)
  - Mapper to Driver
  - Driver to Driver/Mgmt. Agent
- Non-Sequential Control (Regions, e.g.:)
  - Split Requests/Responses
  - Interrupts
- Resource Allocation, e.g.:
  - Dynamic Memory Assignment
  - DMA Linkage

Metalanguage Illustration

Native OS I/O Transfer Request

"udi_if_start" (HPUX)

Mapper

udi_if_start

- **obtain control block (cb)**
- **obtain operating system parameters**
- **build udi/sci parameters in cb**
- **call metalanguage service (udi_sci_io_req)**
- **suspend (non-blocking)**

udi_sci_conf

SCI Driver

udi_sci_io_req

- **obtain udi/sci parameters from cb**
- **setup device for I/O operation**
- **start I/O (prototype involves PCI bridge)**
- **exit**

(call back)

- **put I/O parameters (e.g., status) in cb**
- **call metalanguage service (udi_sci_conf)**

Mgmt Agent

- **Channel Management**
- **"Target" Region Synchronization**
  - **instance management**
  - **callback management**
- **Metalanguage Entry Point Dispatch**

Metalanguage Library Legend

- **SCI Metalanguage**
- **Mgmt Agent Metalanguage**
- **OS Dependent Library**